



# **Intel® Pentium® 4 Processor Identification and the CUID Instruction**

**Revision - 001**

July, 2000



Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel's Intel Architecture processors (e.g., Pentium processor, Pentium® processor with MMX™ technology, Pentium® Pro processor, Pentium® II processor, Pentium® II Xeon™ processor, Pentium® III processor, Pentium® III Xeon™ processor and Intel® Celeron™ processor) may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation  
P.O. Box 7641  
Mt. Prospect IL 60056-7641

or call 1-800-879-4683  
or visit Intel's website at <http://www.intel.com/>  
Copyright © Intel Corporation 2000.

\* Third-party brands and names are the property of their respective owners.



## CONTENTS

	PAGE
<b>1 INTRODUCTION.....</b>	<b>5</b>
<b>2 STANDARD CPUID FUNCTIONS.....</b>	<b>5</b>
2.1 VENDOR-ID AND LARGEST STANDARD FUNCTION # (FUNCTION 0) .....	5
2.2 FEATURE INFORMATION (FUNCTION 1) .....	5
2.3 CACHE DESCRIPTORS (FUNCTION 2).....	8
<b>3 EXTENDED CPUID FUNCTIONS.....</b>	<b>9</b>
3.1 LARGEST EXTENDED FUNCTION # (FUNCTION 80000000H).....	9
3.2 EXTENDED PROCESSOR SIGNATURE AND EXTENDED FEATURE BITS (FUNCTION 80000001H) .....	9
3.3 PROCESSOR NAME / BRAND STRING (FUNCTION 80000002H, 80000003H, 80000004H).....	9
3.3.1 <i>Building the processor name</i> .....	9
3.3.2 <i>Displaying the Processor Name</i> .....	10

## TABLES

	PAGE
TABLE 1 - VENDOR-ID AND LARGEST STANDARD FUNCTION.....	5
TABLE 2 - CPUID FEATURE INFORMATION .....	6
TABLE 3 - PENTIUM® 4 FAMILY BRANDS .....	6
TABLE 4 - FEATURE FLAG DEFINITIONS.....	6
TABLE 5 - LARGEST EXTENDED FUNCTION .....	9



**REVISION HISTORY**

Revision	Revision History	Date
-001	Original Issue.	07/2000



## 1 Introduction

This document provides additional information about the CUID instruction specific to the Intel® Pentium® 4 family of processors. For a detailed description of the P6 family CUID format, please refer to the *Intel Architecture Software Developer's Manual, Volume 2* or Application Note AP-485 *Intel Processor Identification and the CUID Instruction*. Software should use the CUID instruction to identify the Pentium® 4 processors and the features supported by the Pentium® 4 processor.

Two sets of CUID functions are supported in the Pentium® 4 family of processors: **standard functions**, and **extended functions**.

## 2 Standard CUID Functions

The standard CUID functions are those functions supported by previous generations of Intel Architecture 32-bit processors (IA-32). A standard CUID functions is invoked by loading the EAX register with a value of 0, 1, or 2 prior to executing the CUID instruction. The standard functions return results in the general-purpose registers EAX, EBX, ECX, and EDX.

The Pentium® 4 processor does not support function 3 the processor serial number feature introduced by the Pentium® III processor.

### 2.1 Vendor-ID and Largest Standard Function # (Function 0)

On IA-32 processors, when EAX is initialized to a value of 0, the CUID instruction returns the largest standard function number supported by the processor in register EAX (refer to Table 1), and the Vendor-ID string in EBX, EDX, and ECX. Software should validate the Vendor-ID before executing additional CUID functions.

Table 1 - Vendor-ID and Largest Standard Function

	31	23	15	7	0
EAX	Largest standard function number supported				
EBX	u (75h)	n (6Eh)	e (65h)	G (47h)	
EDX	I (49h)	e (65h)	n (6Eh)	i (69h)	
ECX	l (6Ch)	e (65h)	t (74h)	n (6Eh)	

### 2.2 Feature Information (Function 1)

On IA-32 processors, when EAX is initialized to a value of 1, the CUID instruction returns the *Type*, *Family*, *Model*, and *Stepping* value in the EAX register (see Table 2). The *Feature Flags* are returned in the EDX register (see Table 4).

The *Brand Id* was first introduced by the Pentium® III processor, model 8, and is also supported by the Pentium® 4 processor. When function 1 is invoked on the Pentium® 4 processor, the *Brand Id* is returned in EBX[7:0] (see Table 3).

Beginning with the Pentium® 4 processor, the value returned in EBX[31:24] reflects the default APIC ID. This is the value assigned by the hardware when the processor comes out of RESET#. Note that it is possible for software to modify the APIC\_ID but the value returned in EBX[31:24] by function 1 of the CUID instruction remains unchanged, always indicating the default APIC\_ID.



Beginning with the Pentium® 4 processor, the value returned in EBX[15:8] reflects the number of 8-byte chunks flushed by the CLFLUSH instruction. For the Pentium® 4 processor this field returns a value of 8 (64-bytes).

**Table 2 - CPUID Feature Information**

Register Bits	Description
EAX [31:14]	Reserved
EAX [13:12]	Processor Type = 00b
EAX [11:8]	Family = 1111b
EAX [7:4]	Model = xxxb, Initially 0000b
EAX [3:0]	Stepping = xxxb
EBX[31:24]	Default APIC ID
EBX[23:16]	Logical processor count
EBX[15:8]	CLFLUSH chunk count
EBX[7:0]	Brand ID
ECX[31:0]	Reserved
EDX[31:0]	Feature Flags

**Table 3 - Pentium® 4 Family Brands**

EBX [7:0]	Intel Brand
00001000b	Intel® Pentium® 4 processor
00001001b	To Be Announced
00001010b	To Be Announced
00001011b	To Be Announced
00001100b	To Be Announced
00001101b	To Be Announced
00001110b	To Be Announced
00001111b	To Be Announced

**Table 4 - Feature Flag Definitions**

Bit	Descriptions
31:30	<b>Reserved.</b>
29	<b>Automatic Clock Control.</b> A value of 1 indicates the Thermal Monitor automatic thermal control circuit (TCC) is supported by this processor.
28	<b>Reserved.</b>
27	<b>Self-Snoop.</b> A value of 1 indicates this processor contains hardware to maintain a consistent memory image in the presence of aliased memory types.
26	<b>SSE2.</b> A value of 1 indicates the presence of the Streaming SIMD Extensions - 2 (SSE2). A value of 0 indicates the extensions are not present in the processor.
25	<b>SSE.</b> A value of 1 indicates the presence of the Streaming SIMD Extensions (SSE). A value of 0 indicates the extensions are not present in the processor.



24	<b>Fast Save and Restore.</b> A value of 1 indicates that the processor supports the FXSAVE and FXSTOR instructions for fast save and restore of floating point context. A value of 0 indicates the instructions are not supported in the processor.
23	Indicates whether the processor supports the MMX™ technology instruction set and architecture. A value of 1 indicates the presence of the MMX feature. A value of 0 indicates the feature is not present in the processor.
22	<b>ACPI Support.</b> A value of 1 Indicates the processor supports clock controlling by software. This bit indicates the processor supports ACPI registers in MSR space and the associated Temperature LVT entry and interrupt polarity MSR.
21	<b>Debug Trace Store.</b> Indicates that the processor has the ability to write a history of the branch to and from addresses into a memory buffer. A value of 1 indicates the presence of the Debug Trace Store feature. A value of 0 indicates the feature is not present in the processor.
20	<b>Reserved.</b>
19	<b>CLFLUSH Instruction.</b> A value of 1 indicates that the processor supports the CLFLUSH instruction. When this bit is 1, the value returned in EBX[15:8] indicates the number of chunks flushed from the cache.
18	<b>Processor Serial Number.</b> A value of 1 indicates the presence of the Processor Serial Number feature, and that the feature is enabled. A value of 0 indicates the feature is not present in the processor or is currently disabled (refer to the register definition for MSR_MISC_CTL for details on how to disable this feature).  <b>The Pentium® 4 processor family does not support this feature.</b>
17	<b>36-bit Page Size Extension (PSE-36).</b> A value of 1 indicates the presence of the PSE feature. A value of 0 indicates the feature is not present in the processor. The 36-bit page size extension adds four bits to the 4Mbyte-page descriptor.  Indicates whether the processor supports 4-Mbyte pages that are capable of addressing physical memory beyond 4GB. This feature indicates that bits 13-16 of the page directory entry encode the upper four bits of the physical address of the 4-Mbyte page.
16	<b>Page Attribute Table (PAT).</b> A value of 1 indicates the presence of the PAT feature. A value of 0 indicates the feature is not present in the processor.  Indicates whether the processor supports the Page Attribute Table. This feature augments the Memory Type Range Registers (MTRRs), allowing an operating system to specify attributes of memory on a 4k granularity through a linear address.
15	<b>Conditional Move Instruction Supported.</b> The processor supports CMOVcc, and if the FPU feature flag (bit 0) is also set, supports the FCMOVCC and FCOMI instructions.
14	<b>Machine-Check Architecture.</b> A value of 1 indicates the Machine Check Architecture is supported, specifically the MCG_CAP register.
13	<b>Page Global Enable.</b> A value of 1 indicates the global bit in the PDEs and PTEs and the CR4.PGE enable bit are supported.
12	<b>Memory Type Range Registers.</b> A value of 1 indicates the Processor supports the Memory Type Range Registers specifically the MTRR_CAP register.
11	<b>Fast System Call (SEP).</b> A value of 1 indicates the presence of the SEP feature. A value of 0 indicates the feature is not present in the processor.  Indicates whether the processor supports the Fast System Call instructions, SYSENTER and SYSEXIT.
10	<b>Reserved.</b>
9	<b>On-chip APIC Hardware Supported.</b> The processor contains a local APIC.
8	<b>CMPXCHG8 Instruction Supported.</b> The compare and exchange 8 bytes instruction is supported.
7	<b>Machine-Check Exception.</b> Machine Check Exception, INT18, and the CR4.MCE enable bit are supported.
6	<b>Physical Address Extension.</b> Physical addresses greater than 32 bits are supported.



5	<b>Model Specific Registers.</b> Model Specific Registers are implemented with the RDMSR, WRMSR instructions.
4	<b>Time Stamp Counter.</b> The RDTSC instruction is supported including the CR4.TSD bit for access/privilege control.
3	<b>Page Size Extension.</b> The processor supports 4-Mbyte pages.
2	<b>Debugging Extension.</b> The processor supports I/O breakpoints, including the CR4.DE bit for enabling debug extensions and optional trapping of access to the DR4 and DR5 registers.
1	<b>Virtual Mode Extension.</b> The processor supports extensions to virtual-8086 mode.
0	<b>Floating-point Unit On-chip.</b> The processor contains an FPU that supports the Intel387 floating-point instruction set.

### 2.3 Cache Descriptors (Function 2)

When EAX is initialized to a value of 2, the CPUID instruction returns cache and TLB descriptor parameters in the EAX, EBX, ECX and EDX registers.





### 3 Extended CUID Functions

The Pentium® 4 family of processors implement extended CUID functions. To test for support for extended CUID functions software should execute the CUID instruction with EAX = 80000000h. If the CUID instruction returns a value greater than 80000000h then the extended functions are supported.

The following sections describe what the Pentium® 4 processor returns when the allowable extended function numbers are entered into EAX.

#### 3.1 Largest Extended Function # (Function 80000000h)

On the Pentium® 4 family of processors, when EAX is initialized to a value of 80000000h, the CUID instruction returns the largest extended function number supported by the processor in register EAX (refer to 5).

Table 5 - Largest Extended Function

	31	23	15	7	0
EAX	Largest extended function number supported				
EBX	Reserved				
EDX	Reserved				
ECX	Reserved				

#### 3.2 Extended Processor Signature and Extended Feature Bits (Function 80000001h)

On the Pentium® 4 family of processors, when EAX is initialized to a value of 80000001h, the CUID instruction returns 0 in the registers EAX, EBX, ECX and EDX.

#### 3.3 Processor Name / Brand String (Function 80000002h, 80000003h, 80000004h)

Functions 80000002h, 80000003h, and 80000004h each return up to 16 ASCII bytes of the processor name in the registers EAX, EBX, ECX, and EDX. The processor name is constructed by concatenating each 16-byte ASCII string returned by the three functions. The processor name may be right justified and have leading space characters. It is returned in little endian format and NULL terminated. The maximum length of the processor name is 48 bytes. The 48 bytes that comprise the processor name include the *Brand String*, intended speed in ASCII format and the NULL terminator character.

##### 3.3.1 BUILDING THE PROCESSOR NAME

Software must reserve enough space in a byte array to concatenate the three 16 byte ASCII strings that comprise the processor name. Software must execute each function in sequence. After sequentially executing each CUID processor name function, software must concatenate EAX, EBX, ECX, EDX to create the resulting processor name string.

```
Processor_Name    db 48 dup(0)

MOV     EAX, 80000000h
CUID
CMP     EAX, 80000000h    ; Check if extended functions are supported
JB      Not_Supported
MOV     EAX, 80000002h
MOV     DI, OFFSET Processor_Name
CUID    ; Get first 16 bytes of the processor name
CALL    Save_String
```



```
        MOV     EAX, 80000003h
        CPUID                    ; Get second 16 bytes of the processor name
        CALL    Save_String
        MOV     EAX, 80000004h
        CPUID                    ; Get last 16 bytes of the processor name
        CALL    Save_String

Not_Supported:
        RET

Save_String
        MOV     Dword Ptr [DI], EAX
        MOV     Dword Ptr [DI+4], EBX
        MOV     Dword Ptr [DI+8], ECX
        MOV     Dword Ptr [DI+12], EDX
        ADD     DI, 16
        RET
```

### 3.3.2 DISPLAYING THE PROCESSOR NAME

The processor name may be a right justified string padded with leading space characters. When displaying the processor name string the display software must skip the leading space characters and discontinue printing characters when the NULL character is encountered.

```
        CLD
        MOV     SI, OFFSET Processor_Name    ; Point SI to the name string

Spaces:
        LODSB
        CMP     AL, ' '
        JE      Spaces
        CMP     AL, 0
        JE      Done
        JNE     Display_Char

Display_Char:
        CALL    Display_Character           ; Put a char on the output device
        LODSB
        CMP     AL, 0
        JNE     Display_Char
        JNE     Display_Char

Done:
```